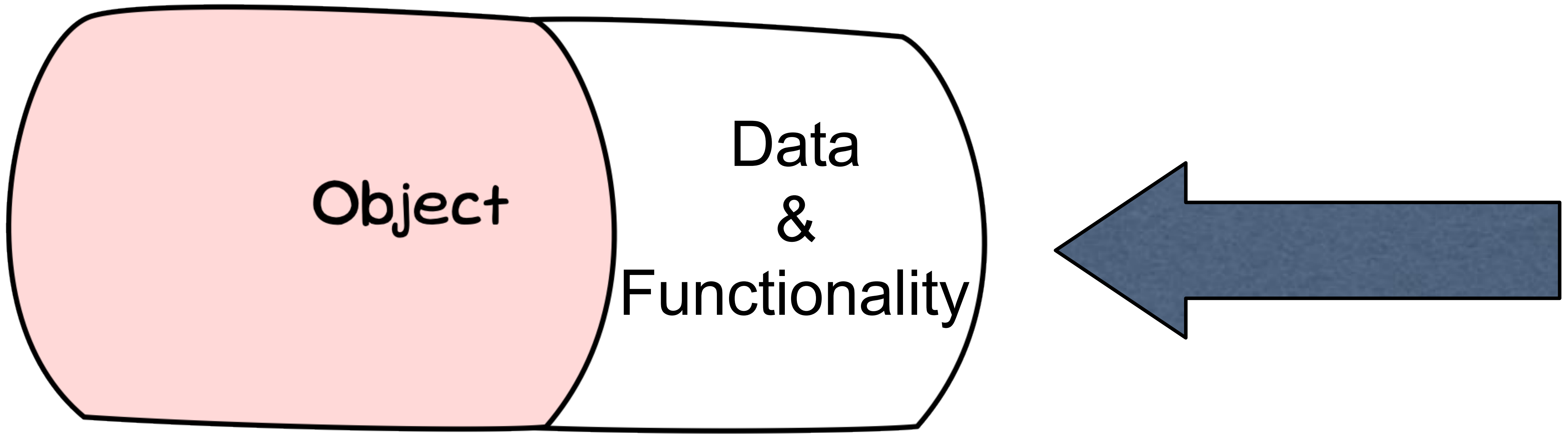
COS20007
Object-Oriented Programming

Topic 05 Part A
Interfaces

Learning Outcomes

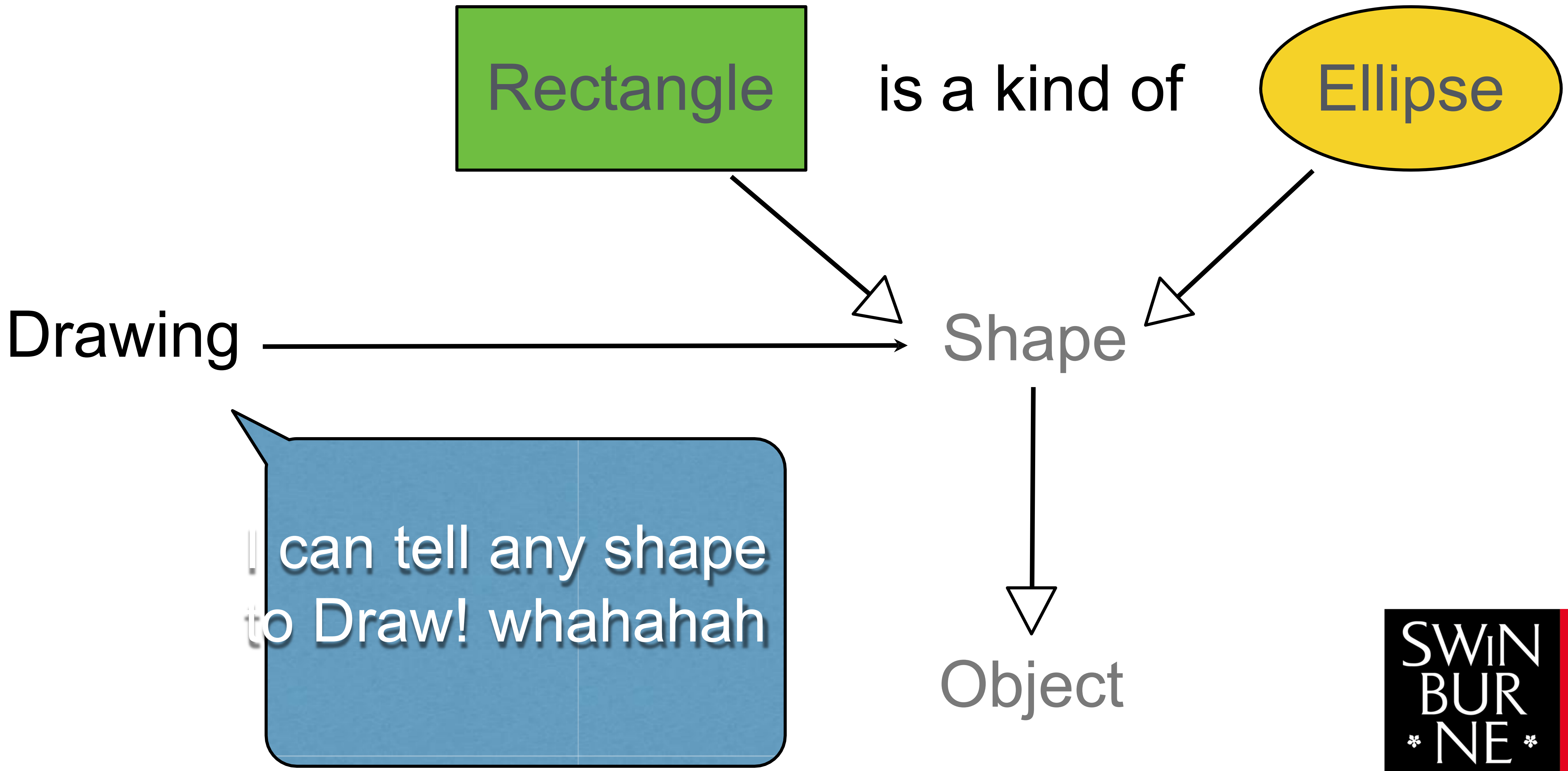
- The importance of interfaces in OOP
- Understand how to implement interfaces
- Demonstrate interfaces with real-world examples

Object oriented programs contain objects that know and can do things



Know things
&
Do things

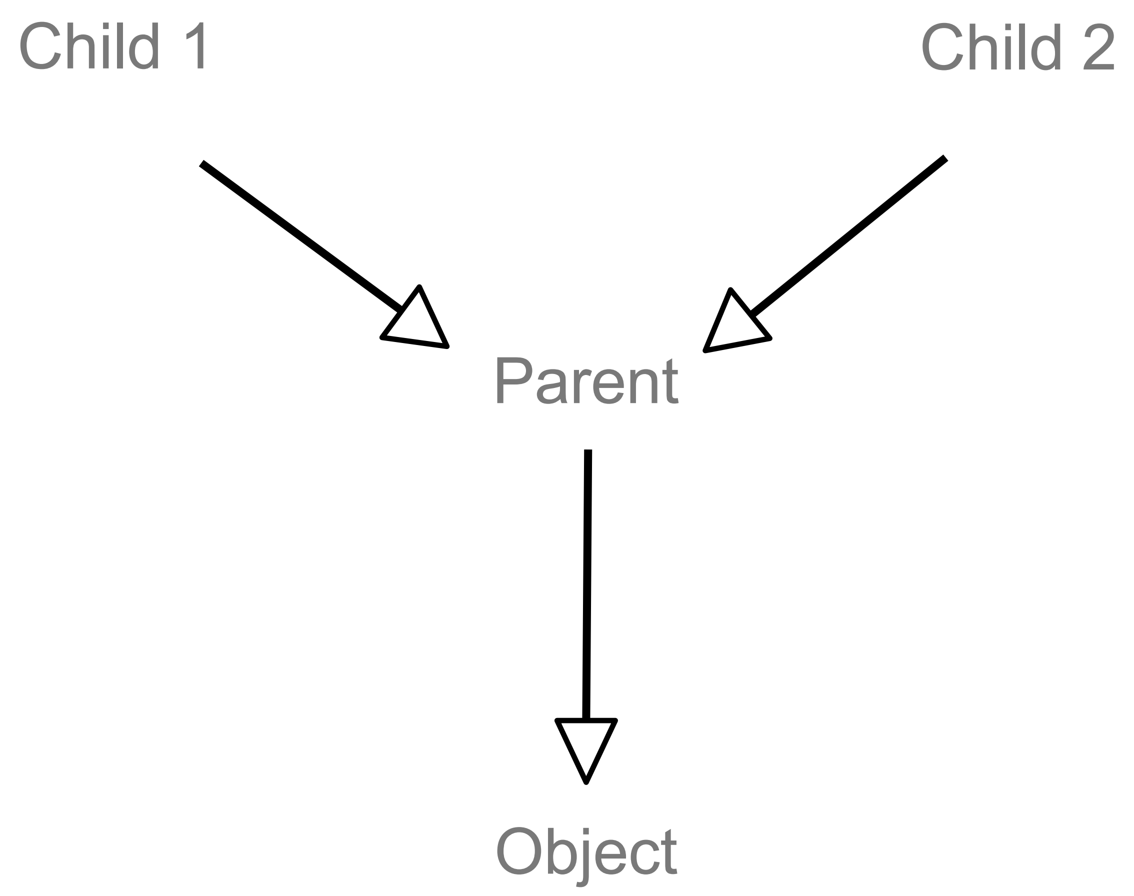
Developers use inheritance to create families of types with common features



What about cases where an object wants to interact, but not with a family of related types

Sorter

I want to sort ... as many kinds as possible...



Ideally the object should be able to say what features they need…

Sorter

I can sort anything with these features (anything Comparable)

Something Comparable must be able to…

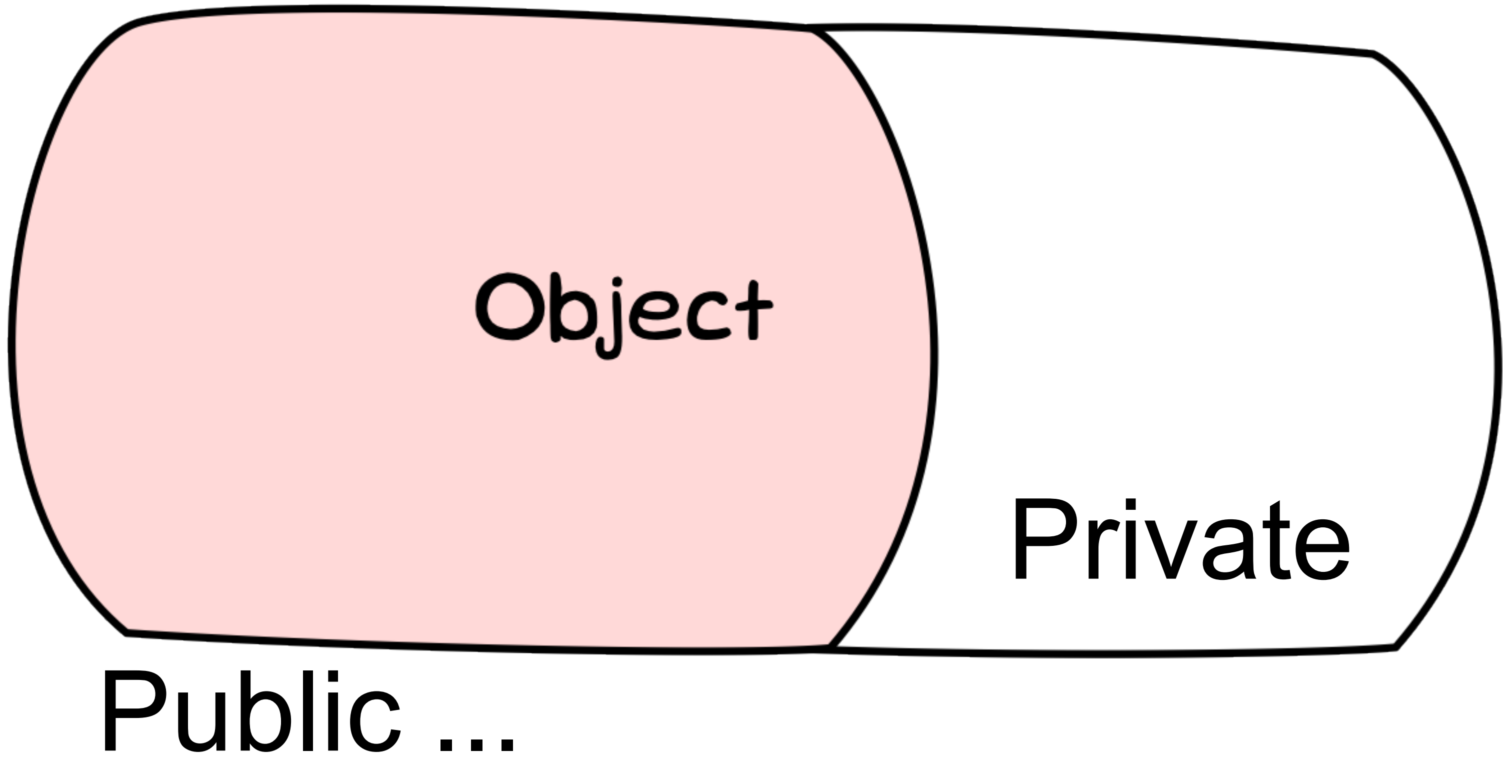
Compare itself **To** another object

Use an interface to define the features you need

SWINBURNE UNIVERSITY OF TECHNOLOGY

Specify the features that implementing classes must provide

To be Comparable you must have an "int Compare(...)" method...

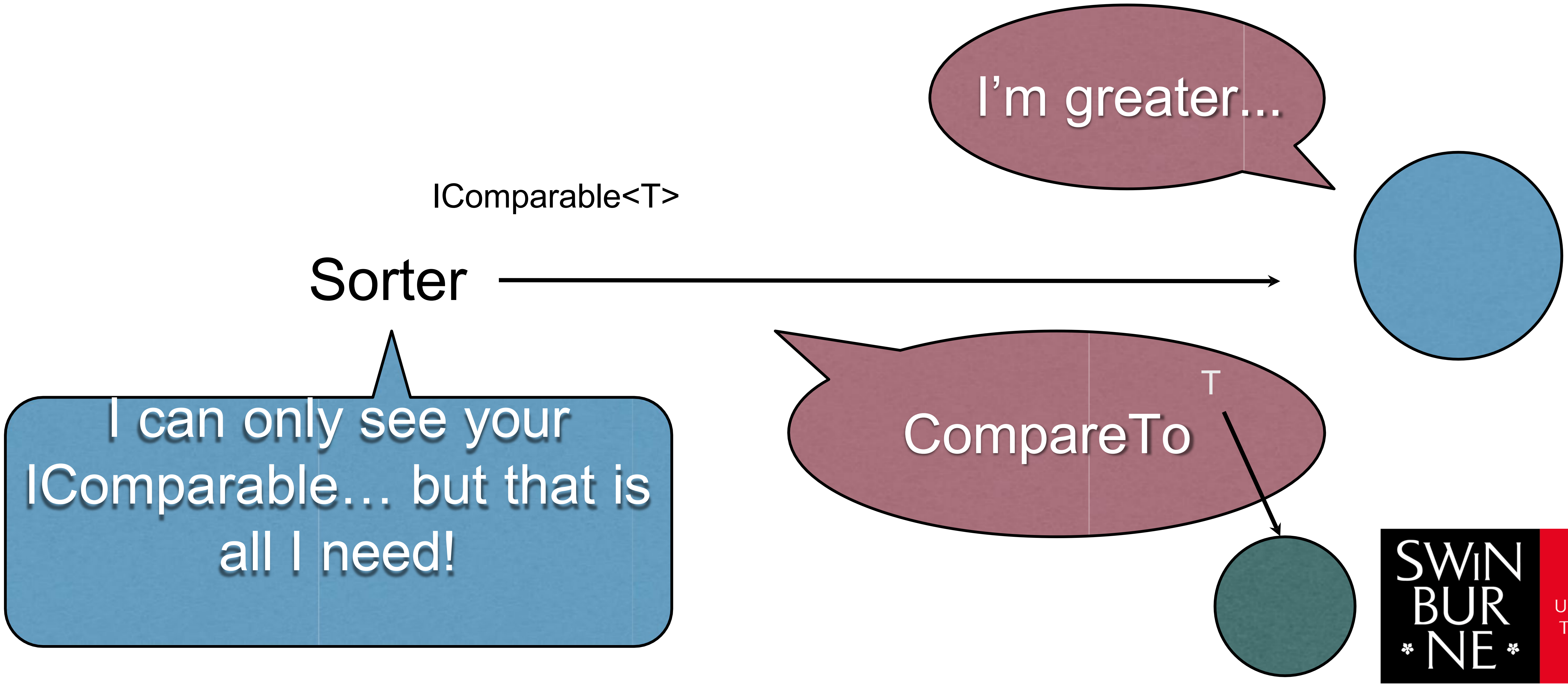


List these required features in the Interface declaration

```
public interface IComparable<in T>
{
    int Compare(T other);
}
```

C# uses an I prefix to interface names.

Use the interface and access these features on whatever is supplied to you!



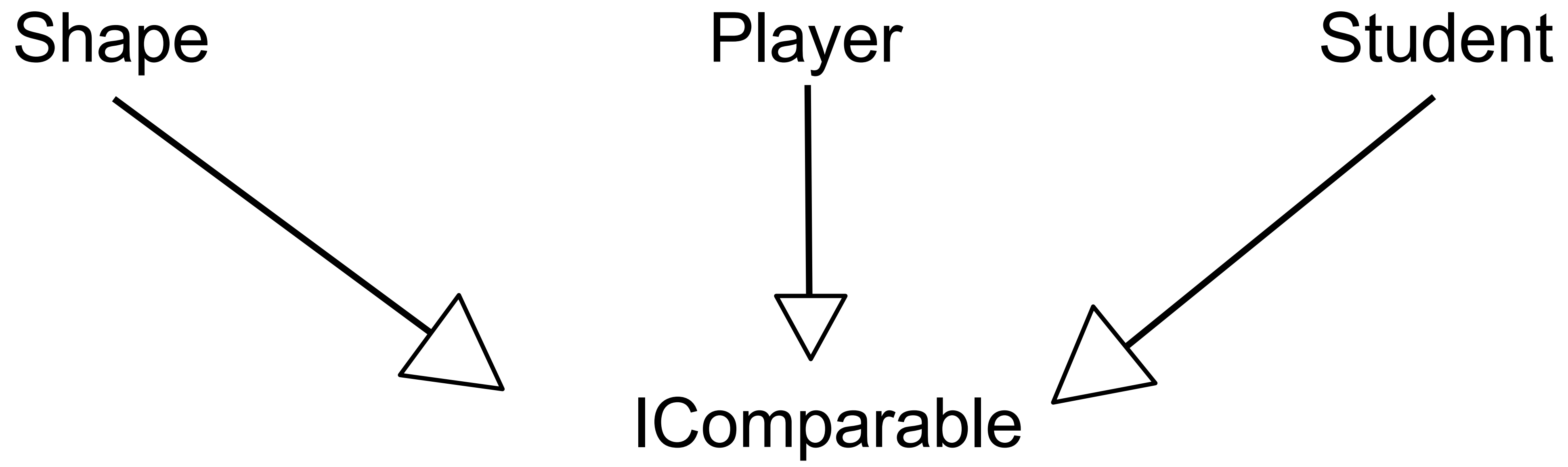
Implement the interface if you want the services provided

Implement the interface for any role that wants to be used by the other object

```
public class Student : IComparable<Student>
{
    public int Compare(Student other) {...}
}
```

```
     Student student1 = new Student();
     Student student2 = new Student();
int result = student1.Compare(student2);
```

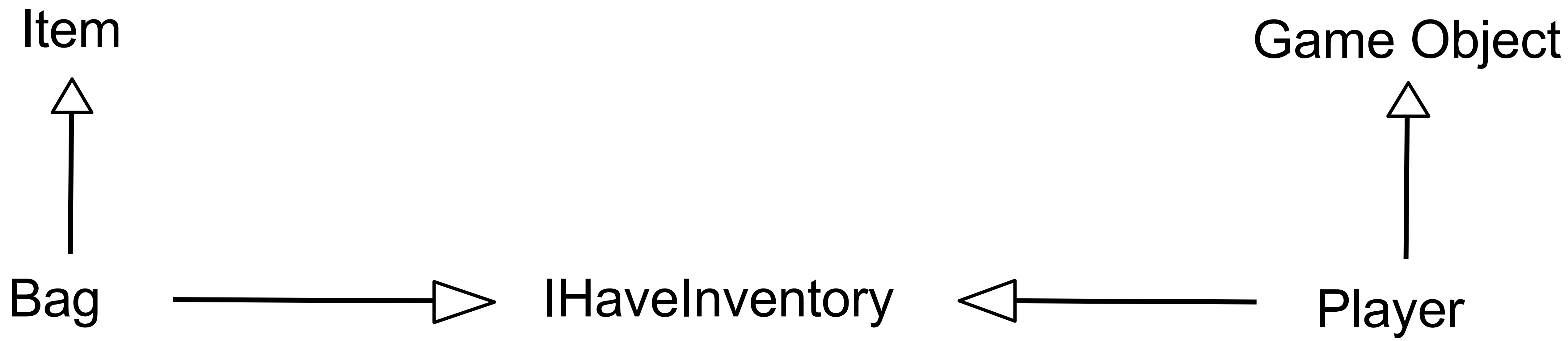
Polymorphism means objects of this type can now be used anywhere the interface is needed



```
List<IComparable> myGenericList = new List< IComparable >();
Shape myShape = new Shape();
Player myPlayer = new Player();
myGenericList.Add(myShape);
```

Classes can inherit from **one** class, but can implement **many** interfaces

Example 1.



Example 2. `public class Square : IComparable<Square>, IPrintable<Square>`

- Microsoft C# interfaces, https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/interfaces

Take away message

- Standard inheritance is only used when we have a family of related types only.

- We use interfaces to define the features we need when a family of types does not make sense

- A class can implement many interfaces

- Interfaces allow you to access features in a flexible way with polymorphism

SWINBURNE UNIVERSITY OF TECHNOLOGY